Web Services for System i

Dan Shinedling, Jr.
KS2 Technologies, Inc.
Metro Midrange Systems Association
November 21, 2006

Agenda

- Introducing Web Services
- Creating a Web Service with WDSC
- Demonstration
- Practical Examples
- Summary

About the Speaker

- Dan Shinedling, Jr., President of KS2 Technologies, Inc., 11 years
- IBM Certified Expert – System i
- KS2 is a Premier IBM Business Partner, focus on:
  - System i sales installation and support
  - System i Integration and Conversions
  - Oracle Applications (JDE, E1, World)
  - Solutions for Homebuilding Industry
- My personal focus - Getting the most out of the System i, exploiting new technology and enhancements for System i:
  - Nineties – PC Support, CA Windows, TCP/IP, TN5250, Automated FTP's, RPGLE
  - 2000's – E RPG, eXLerators™, WebSphere, WDSC, Web Services, Java
- Went live on first System i Web Service in May 2006.
- Contact – danjr@ks2inc.com, 817-416-5505 x 205

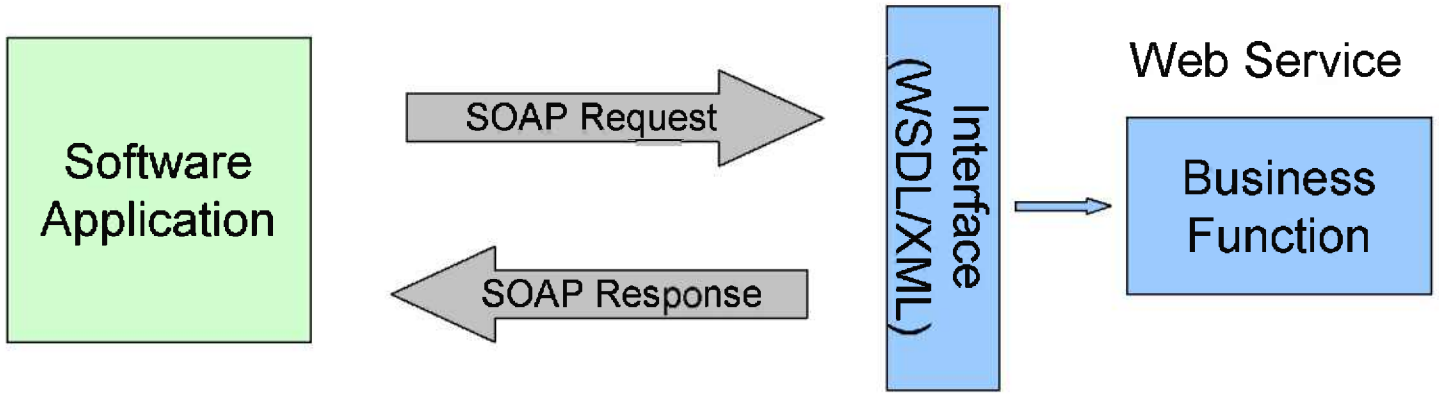
Introducing Web Services

- Web services are self contained applications with a well defined, published, and platform independent interface.
- The language of the Web Service is XML.
- Web services use a standard protocol to describe operations that can be performed or data that can be exchanged

Terms

- XML – Extensible Markup Language. Generic language used to describe any kind of content in a structured way, separated from its presentation to a specific device
- SOAP – Simple Object Access Protocol. Platform neutral protocol that allows a client to call a remote service. Message format is XML.
- WSDL – Web Services Description Language, also in XML. Describes the Web Service:
  - The operations that a Web Service performs
  - The parameters and data types for these operations
  - Service access info: location (IP address, port), protocol (HTTP, SMTP…)
- PCML – Program Call Markup Language. Used by iSeries Java toolbox to generate program calls to native RPG program.

Schematic


Software Application
SOAP Request
SOAP Response
Interface (WSDL/XML)
Web Service
Business Function

Advantages

- Native System i programs are instantly available to non-native applications
- Web Services provide message level or transaction level feedback than can be used by the calling application
- XSLT, Extensible Stylesheet Language Transformations, can be used to “translate” messages between different services to make them compatible, without changing code.

Today's Demonstration

- Use WDSC Wizard to create a Web Service from an existing RPG program
- Use the wizard to create a test client to test the Web Service
- Deploy the Web Service to a WebSphere Application Server running on System i
- Practical Examples

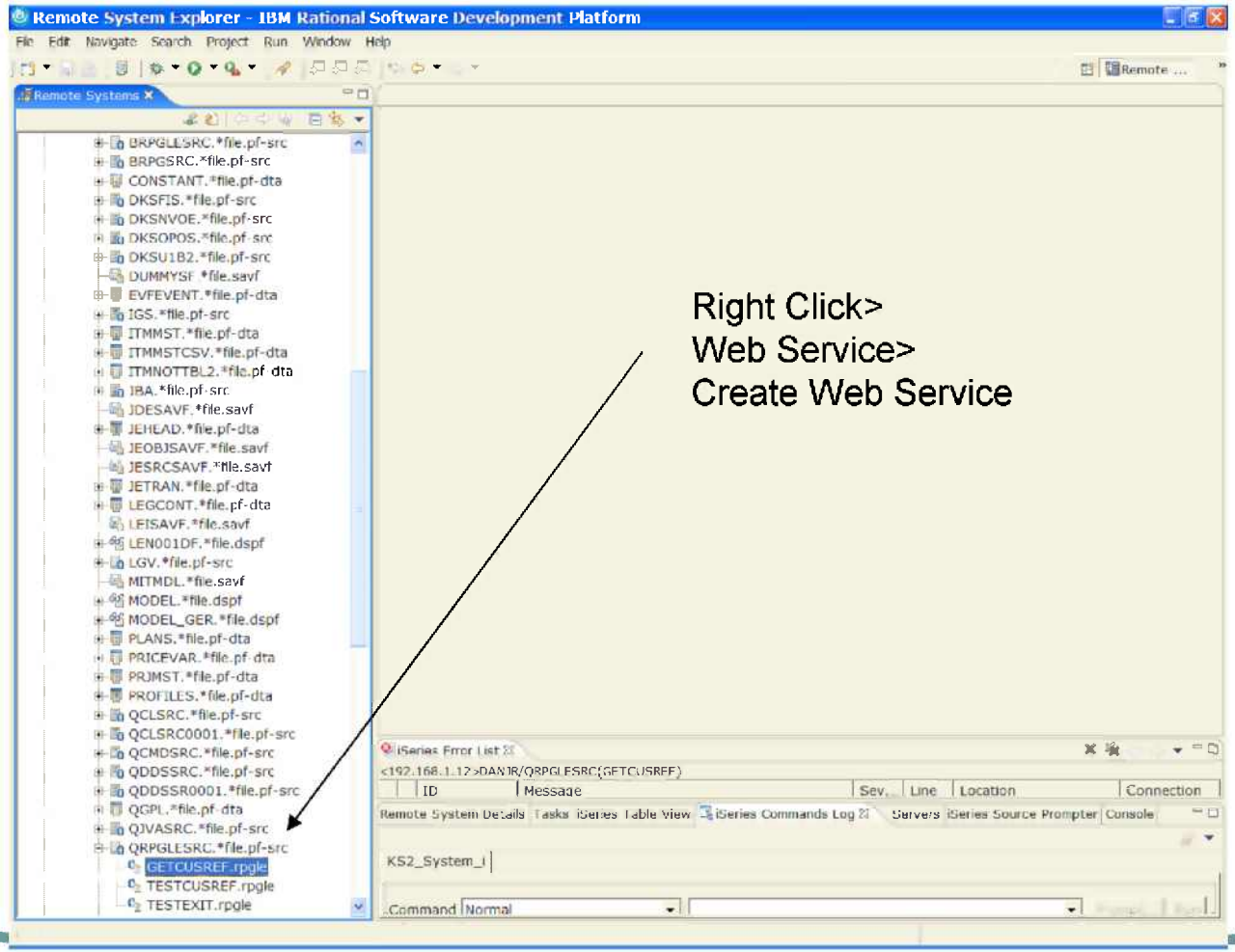
System Recommendations

- Rational Software Development Platform version 6.0.1 (WDSC)
- Used IBM Rational Product Updater to get most recent updates… NOTE – FTP download did not work, took about 6 hours to run. Don't waste your time unless you've updated the product
- System i, V5R3
- WebSphere Application Server V6 Express

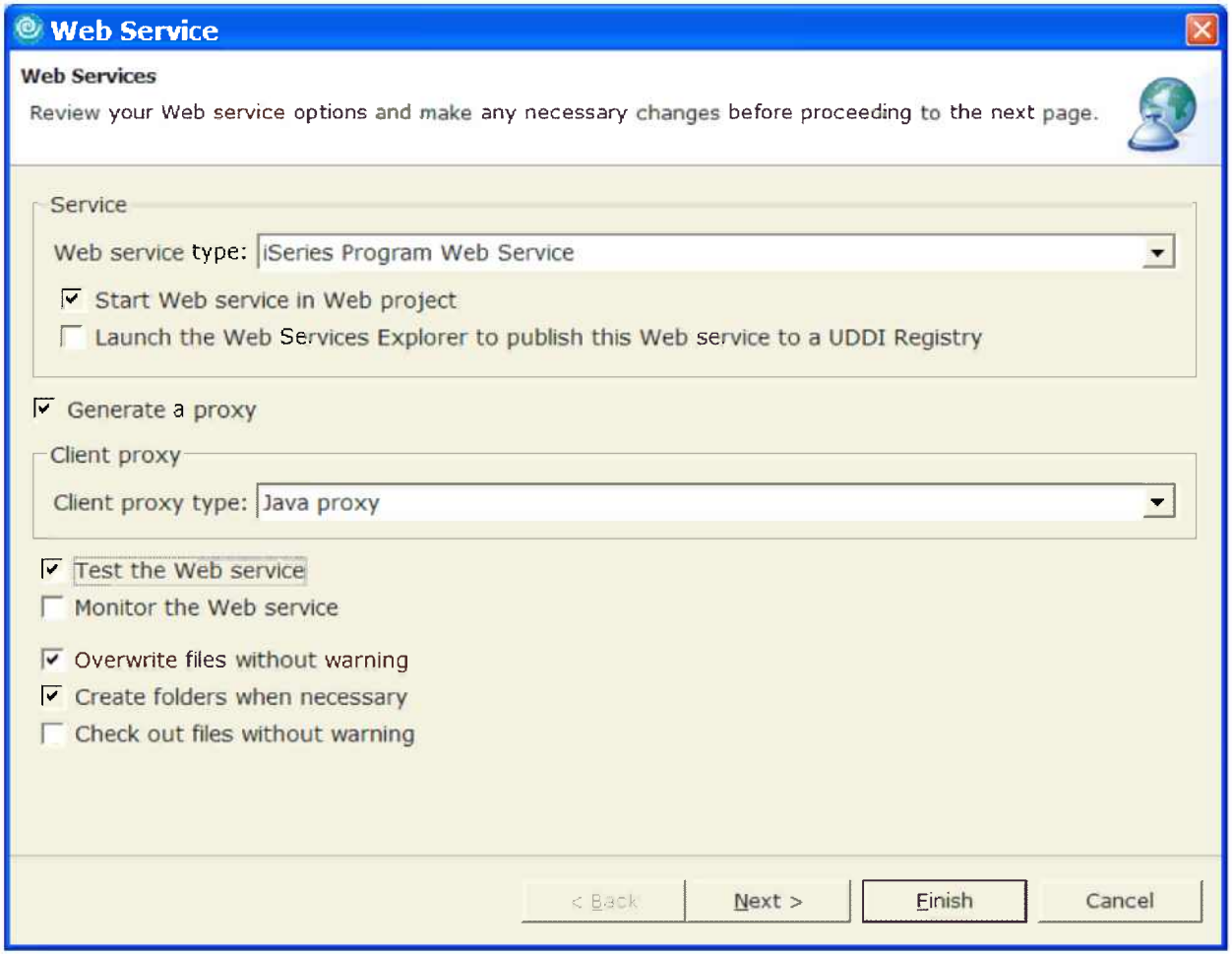
Step 1 – Use Wizard to Create Web Service

1. Remote System Explorer
2. Add Library to Library List by right clicking library
3. Right click SOURCE member
4. Web Service > Create Web Service (starts wizard)
5. Leave defaults, check Generate Proxy and Test Web Service (these are for testing only).

Select Source Member



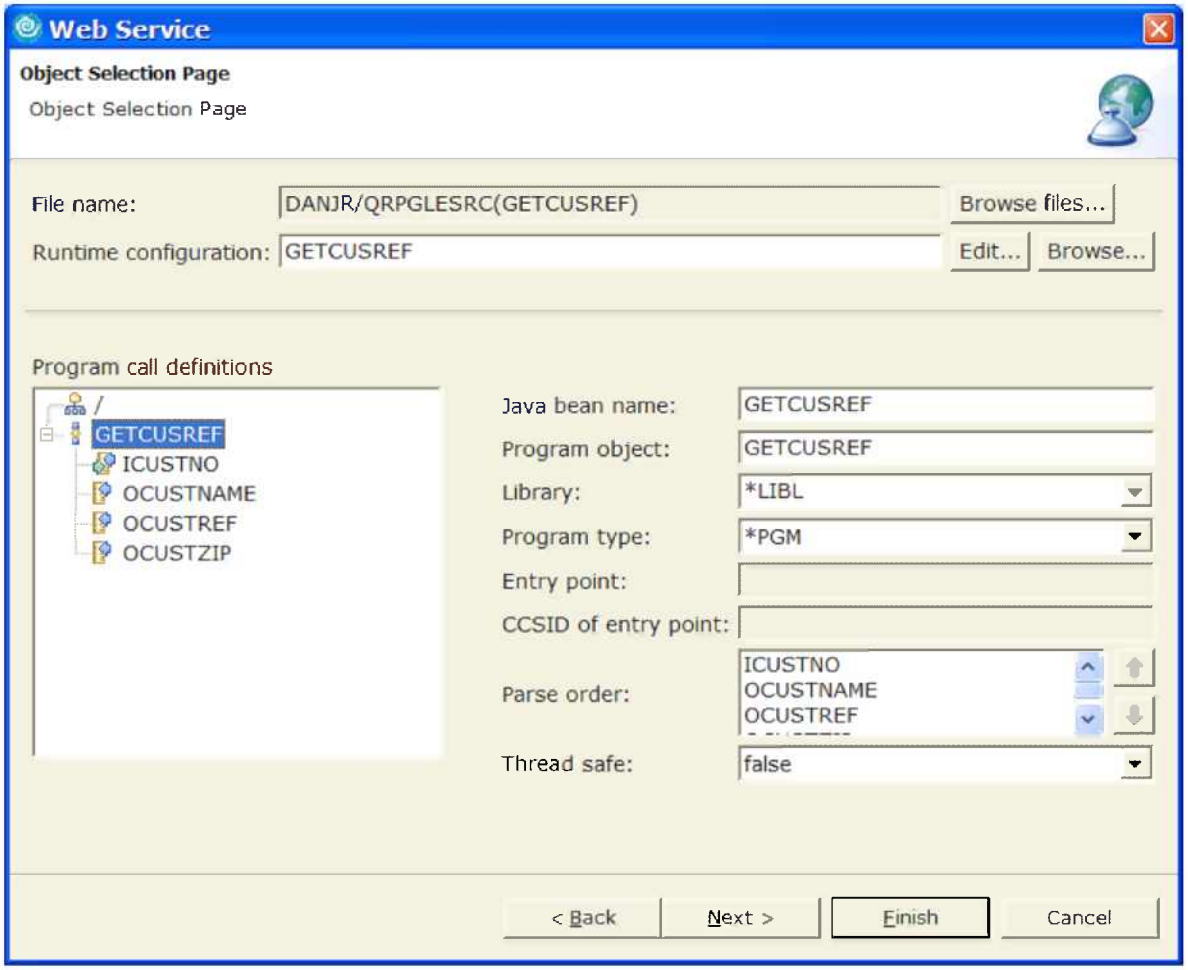
Screen Shot


Web Service
Web Services
Review your Web service options and make any necessary changes before proceeding to the next page.
Service
Web service type: iSeries Program Web Service
Start Web service in Web project
Launch the Web Services Explorer to publish this Web service to a UDDI Registry
Generate a proxy
Client proxy
Client proxy type: Java proxy
Test the Web service
Monitor the Web service
Overwrite files without warning
Create folders when necessary
Check out files without warning
< Back
Next >
Finish
Cancel

Object Selection Page

1. EDIT – to specify runtime user and lib list
2. Program definitions:
    1. Program Type - *SRVPGM or *PGM. GOTCHA: THIS WILL GET SCREWED UP so be careful. Check PCML when wizard is done!
    2. Define PARMS as Input, Output or Both. By default, they are probably both.
3. Go back and check program type

Screen Shot


Web Service
Object Selection Page
Object Selection Page
File name:
DANJR/QRPGLESRC(GETCUSREF)
Browse files...
Runtime configuration:
GETCUSREF
Edit...
Browse...
Program call definitions
/
GETCUSREF
ICUSTNO
OCUSTNAME
OCUSTREF
OCUSTZIP
Java bean name:
GETCUSREF
Program object:
GETCUSREF
Library:
*LIBL
Program type:
*PGM
Entry point:
CCSID of entry point:
Parse order:
ICUSTNO
OCUSTNAME
OCUSTREF
Thread safe:
false
< Back
Next >
Finish
Cancel